Accessing files from the smallest devices to the largest (and the cloud): Improvements to SMB3.1.1 and Linux

Presented by Steve French
Principal Software Engineer
Microsoft Azure Storage
- This work represents the views of the author(s) and does not necessarily reflect the views of Microsoft Corporation.
- Linux is a registered trademark of Linus Torvalds.
- Other company, product, and service names may be trademarks or service marks of others.
Who am I?

- Steve French smfrench@gmail.com
- Author and maintainer of Linux cifs vfs (for accessing Samba, Azure, Windows and various SMB3/CIFS based NAS appliances)
- Co-maintainer of the new kernel server (ksmbd)
- Also wrote initial SMB2 kernel client prototype
- Member of the Samba team
- Coauthor of SNIA CIFS Technical Reference, former SNIA CIFS Working Group chair
- Principal Software Engineer, Azure Storage: Microsoft
Outline

- Overview of Linux FS activity
- New Kernel server
- Recent client improvements
  - Kernel
  - Utilities
- Coming soon … what to look forward to
- Testing improvements
A year ago and now ...

- Now: 5.19-rc1 “Superb Owl”
- Then: 5.13-rc2 “Frozen Wasteland”
LSF/MM/eBPF summit recently concluded
Some Linux FS topics of interest from LSF and other recent discussions

- Folios, netfs and the redesign of page cache and offline (fscache)
- Improvements to statx and fsinfo and to inotify/fanotify
- Idmapped mounts
- Extending in kernel encryption: TLS handshake (for NFS) and QUIC (SMB3.1.1 and other)
- io_uring (async i/o improvements)
- Shift to cloud
- Better support for faster storage (NVME) and net (RDMA/smbdirect)
Linux Filesystems Activity over past year (since 5.13-rc2)

- 5320 filesystems changesets (6.6% of total kernel changesets, one of the most watched parts of the kernel, and FS Activity up slightly as percentage of kernel activity)
- Linux kernel fs are 1.07 million lines of code (measured this week)
Most Active Linux Filesystems over the past year

- VFS (mapping layer) 1022 changesets
- The top filesystems and VFS dominate the activity
- Most active are BTRFS 927 (almost 60KLOC changed!), XFS 519
- Then NFS (283) and SMB3.1.1 (cifs.ko) (273 changesets) clients
  - cifs.ko had almost twice as many lines changed. It has been a VERY active year for cifs.ko
- Other:
  - Ext4 (263 changesets), ksmbd (new, added in the 5.15 kernel) (189), nfsd (177), ceph (139), gfs2 (133), ntfs3 (new, added in 5.15) (123)
SMB3.1.1 Activity was strong this year

- cifs.ko activity was strong, 273 changesets
  - cifs is now 60KLOC kernel code (not counting user space utilities), 2% larger

- ksmbd activity was also strong
  - Introduced in the 5.15 kernel, 25KLOC kernel code, 189 changesets since its introduction

- Samba server (userspace) is over 3.5 million lines of code (orders of magnitude bigger than the kernel smbd server or any of the NFS servers) and is even more active
One of the strengths of SMB3.1.1 is broad interop testing

- Hopefully in-person plugfests resume soon – but a virtual SMB3.1.1 plugfest will be held this week
What are the goals?

- Repeating an older slide about goals of SMB3.1.1:
  - Fastest, most secure general-purpose way to access file data, whether in the cloud or on premises or virtualized
  - Implement all reasonable Linux/POSIX features – so apps don’t know they run on SMB3 mounts (vs. local)
  - As Linux evolves, and needs new features, quickly add to Linux kernel client and Samba and ksmbd
Progress and Status update for Linux Kernel Server (ksmbd)

Provided by Namjae Jeon (linkinjeon@kernel.org)
ksmbd merged into mainline Linux in the fall (5.15)

• Getting reviews for 5 months since ksmbd v1 patch series went in Linux-next
• Many high profile developers reviewed, Thank you!
  • Multiple security issues were recently identified, and these fixes are the focus now
• Ksmbd is merged into linux-5.15-rc1 merge window
• To make module and directory name consistent: changed “cifsd” to “ksmbd”
• Later the cifs source directory will be renamed to smbfs_client to reduce confusion (and to avoid referencing old, deprecated, less secure protocol dialect ‘cifs.’ Modern clients and servers negotiate SMB3 or later, not old cifs)
  • Common code between client and server is now in “fs/smbfs_common” directory
AES-256 encryption support

• ksmbd AES-256 CCM/GCM encryption support now available (strongest encryption)
• Ksmbd accelerated encryption (AES-GCM) performance using AES-NI support in kernel
Kerberos support

• Support authentication with Kerberos
• Ksmbd transmit Kerberos msg to ksmbd.mountd
• Ksmbd.mountd uses libkrb5 library
Duplicate extent support

- Ksmbd add support for FSCTL_DUPLICATE_EXTENT_TO_FILE
- This command can be used if share is in reflink support local fs (Linux client uses it for some fallocate related operations like insert range)
- Additional xfstests tests pass.
- Ksmbd doesn’t have to deal with VFS mapping(btrfs, etc.) layer like samba.
SMB3 multi-channel support

• SMB3 Multichannel feature greatly improves performance on Multi-port NIC or multiple NICs.
• Ksmbd kernel server started to support SMB3 multichannel.
• TODO Replay/retry features on channel failure.
SMB3 multi-channel support

• Send NICs information to client through FSCTL_QUERY_NETWORK_INTERFACE_INFO command
SMB3 multi-channel support

- Client send session binding request to ksmbd.
SMB3 multi-channel support

• Client send interleaved write requests to dual channels(192.168.0.3, 192.168.0.4)
Currently working features

- SMB Direct with windows client
  - Got test HW support from Chelsio (Bob Dugan)
  - Windows client connection success
  - Checking performance issue
  - Add interface to change SMBD parameter as per RDMA NICs
  - Credit management rework

- SMB2 directory leases

- SMB2 change notify
  - Considering using fanotify instead of inotify for SMB2\_WATCH\_TREE
  - Need to change fanotify codes as export symbol to call function by ksmbd.
ksmbd start to fully support smbdirect

• Handle large RDMA read/write size(bulk data) supported by SMB Direct multi-descriptors.(It supported single descriptor with 512KB size before)
  • 8MB RDMA read/write size by default.
  • Control read/write size through ksmbd configuration.(e.g. smbd io size = 16MB)

• Improve the compatibility with various RDMA types of NICs
  • Tested smb-direct working with iWARP(Chelsio, soft-iWARP), Infiniband(Mellanox NICs, Connectx3 ~ x5), ROCE(soft-ROCE).

• Auto-detection of RDMA NICs without configuration.
  • Server should send RDMA NIC info to client.
  • No need to specify RDMA NIC information to smb.conf.
Performance test environment

- Benchmark tool

- Server
  
  CPU: intel Silver 4114 x 2
  DRM: 512GB
  NVMe SSD: Kioxia CM6 1.9T x 9(mdadm raid0 with XFS)
  NIC: MCX516A-CCAT

- Client
  
  CPU: intel Silver 4215 x 2
  DRM: 64GB
  NIC: MCX516A-CCAT
Performance comparison between single and multi-descriptor.

Frametest –w 4k –n 2000 –t 20
Frametest –r 4k –n 2000 –t 20

Write speed avg
Read speed avg

single desc  multi-desc
RDMA write performance per number of connections

Frametest –w 4k –n 2000 –t 20

Number of RDMA Connections

RDMA write size
RDMA read performance per number of connections

Frametest –r 4k –n 2000 –t 20

Number of RDMA Connections

RDMA read size

- 8MB
- 16MB
If interested in contributing there are lots of cool features to work on, as well as improved integration with Samba (e.g. user space upcalls for additional features). The SMB3.1.1 family of protocols is huge!

- Roles: there are multiple developers helping Namjae (the maintainer). I am managing the git merges, ensuring additional functional testing is done regularly, and reviewing patches as requested by Namjae (my focus is largely on the client)
- Namjae would welcome additional help with code reviews, security auditing, testing and new features
- Very exciting time!
Recent improvements in the kernel client
(cifs.ko)
AES-GCM-256 (Strongest Encryption)

- Negotiated by default if server requires it (Azure, Windows, ksmbd etc. support it now)
- Client can require (force) if “require_gcm_256” module parm set
- See trace of Linux AES-GCM-256 mount to Windows (with “require_gcm_256” set on client)
It rocks! Let’s take a simple example and copy 10GB from Azure server down to Linux client VM
   “dd if=/mnt/10GB of=/dev/null bs=1M count=10K”
- Old defaults (3.0) 143MB/sec
- With 3.1.1 201MB/sec (41% faster)
- And go to 2 channels & set new parm “rasize” to 4MB
453MB/sec, More than 3x faster!!
- Lots of great perf improvements!
Support added for handle leases (deferred close) in 5.13 kernel. Here are two simple example of the huge caching perf gains even copying to Samba localhost

- Create a 2GB file and read it back (read is 4x faster)

```
dd if=/dev/urandom of=2G bs=1M count=2K ;
dd if=2G bs=1M count=2K of=/dev/null
```

- Before: 2.0 GiB copied, 0.583143 s, 3.7 GB/s
- Current: 2.0 GiB copied, 0.159237 s, 13.5 GB/s

- Read the same 4GB file twice (2\textsuperscript{nd} time is 3x faster)

```
dd if=4G of=/dev/null bs=1M count=4K ;
dd if=4G of=/dev/null bs=1M count=4K
```

- Before: 4.0 GiB copied, 1.36794 s, 3.1 GB/s
- Current: 4.0 GiB copied, 0.441635 s, 9.7 GB/s
5.13 kernel (June 27th 2021)
66 changesets. cifs.ko version 2.32

- Huge performance boost for readahead in some configurations by setting new mount parameter ("rsize=") larger than rsize
- Add support for fcollapse and finsert (collapse and insert range calls)
- Add support for deferred close (handle leases), greatly improving performance of some workloads
- Improvements to directory caching of the root directory
- Strongest type of encryption (GCM256) is now sent by default in the list of allowed encryption algorithms (GCM128 preferred, then GCM256 then CCM128) and does not have to be enabled manually in module load time parameters
- Debugging of encrypted mounts improved (e.g. for multiuser mounts and also for GCM256)
- Add support for shutdown ioctl (useful to halt new activity to better allow emergency unmounts, and also required for some common testcases)
- Mount error handling improvements (see "/proc/fs/cifs/mount_params"
5.14 kernel (August 29\textsuperscript{th}) 71 changesets, cifs.ko version 2.33

- Falloate improvements (can now alloc smaller ranges up to 1MB). Thank you Ronnie!
- DFS reconnect improvements, and reconnect retry improvements. Thank you Paulo!
- Experimental support added for negotiating signing algorithm
- And 5.15 kernel
  - Important deferred close (handle lease) bug fixes
  - Support for weaker authentication (NTLMv1 and LANMAN) removed
  - (And experimental kernel server, ksmbd, merged)
5.16 kernel (Jan 9, 2022) 46 changesets
cifs.ko version 2.34

- Performance improvements for stat, setfilesize and set_file_info (additional uses of compounding)
- Multichannel improvements (thanks Shyam!)
- Reconnect improvements
- Fscache fixes
- New mount parm “tcpnodelay”
5.17 kernel (March 20th), 51 changesets, cifs.ko ver 2.35

- Add support for new fscache (offline files caching mechanism)
- Send additional NTLMSSP info (including module and OS version) for improved debugging
- DFS and ACL fixes
- Restructuring of multichannel code
Important performance improvement (reuse cached file handle for various common operations like stat and statfs if available), greatly reducing metadata operations (like open/close)

Important fscache (offline file caching) and DFS improvements

cross mount reflink now supported, which can dramatically improve copy performance from one share to another (on the same server) if they support duplicate extents.
5.19-rc kernel (expected in early August)

- Important performance optimization for directory searches, now we cache the root directory content (to the many servers which support directory leases) reducing amount of network traffic for queries in the root directory
- Multichannel reconnect improvements (e.g. when address or interfaces change)
- RDMA (smbdirect) improvements
Tracing continues to improve ...

- Added more than 10 additional dynamic tracepoints
eBPF is amazing ...

- See Brendan Gregg’s website
- Also see e.g. [https://wiki.samba.org/index.php/LinuxCIFS_troubleshooting](https://wiki.samba.org/index.php/LinuxCIFS_troubleshooting)
- Can be as simple to do as “trace-cmd record -e cifs”
  - And then “trace-cmd show” in another window
- Let us know if suggestions on other debugging tracepoints that would be helpful
- And don’t forget about proc/fs/cifs/Stats, proc/fs/cifs/open_files and proc/fs/cifs/DebugData ...
Recent improvements – cifs-utils

Userspace tools
Improved user space tools (cifs-utils)

- cifs-utils 6.14 released in Sept
  - Add commands to view Alternate Data Streams
  - setcifsacl improvements
  - Improved debugging (keydump)

- Cifs-utils 6.15 released in April

- More recently
  - Add support for gss-proxy (improving krb5 credential retrieval)
  - Misc. bug fixes
eBPF is amazing ...

- See Brendan Gregg’s website
- Also see e.g. [https://wiki.samba.org/index.php/LinuxCIFS_troubleshooting](https://wiki.samba.org/index.php/LinuxCIFS_troubleshooting)
- Can be as simple to do as “trace-cmd record -e cifs”
  - And then “trace-cmd show” in another window
- Let us know if suggestions on other debugging tracepoints that would be helpful
- And don’t forget about `proc/fs/cifs/Stats`, `proc/fs/cifs/open_files` and `proc/fs/cifs/DebugData` ...
Coming soon ...

New features under development for SMB3.1.1 on Linux
What features can you expect in next few releases?

- Extending use of directory leases to greatly improve metadata caching (currently limited to contents of the root directory)
- SMB3.1.1 compression support (allow compressing network traffic based on the SMB3.1.1 compress mount parm)
- statx to return additional SMB3.1.1 attributes like “offline”
- Improvements to enable fanotify/inotify over SMB3.1.1 mounts (currently requires a private SMB3.1.1 specific ioctl)
- Prototype of SMB3.1.1 over QUIC (new encrypted network transport)
- More perf improvements for folios, cache, parallel i/o, multichannel
- More testing of the SMB3.1.1 POSIX extensions to ksmbd (and soon Samba server)
Testing Improvements

Section Subtitle
Automated testing has greatly improved

- Historically SMB3.1.1 plugfests multiple times a year have helped too
- The ‘buildbot’ continues to improve, more tests added, reducing regressions and improving quality
- Test groups for different server types and a general “cifs-testing” one
Additional tests are encouraged

- Xfstests are the standard Linux filesystem functional tests
- Added 21 to the main “cifs-testing” regression testing group (up to 245 tests run on every checkin from this group)
- Various server specific groups have added even more
  - Azure SMB3.1.1 multichannel: up 25% more tests, now includes 133 tests
  - Ksmbd (Linux kernel server target) up 15%, now includes 144 tests
- There are detailed wiki pages on wiki.samba.org going through how to setup xfstests with cifs.ko, and what features need to be added to enable more tests (tests that currently skip or fail so aren’t run in the ‘buildbot’)
Thank you for your time

- Future is very bright!
Additional Resources to Explore for SMB3 and Linux

- Samba-technical mailing list and IRC channel
- And various presentations at [http://www.sambaxp.org](http://www.sambaxp.org) and Microsoft channel 9 and of course SNIA … [http://www.snia.org/events/storage-developer](http://www.snia.org/events/storage-developer)
- And the code:
  - [https://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/tree/fs/cifs](https://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/tree/fs/cifs)
  - For pending changes, soon to go into upstream kernel see:
    - [https://git.samba.org/?p=sfrench/cifs-2.6.git;a=shortlog;h=refs/heads/for-next](https://git.samba.org/?p=sfrench/cifs-2.6.git;a=shortlog;h=refs/heads/for-next)
  - Kernel server code: [https://git.samba.org/ksmbd.git/?p=ksmbd.git](https://git.samba.org/ksmbd.git/?p=ksmbd.git) (ksmbd-for-next branch)